(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 January 2001 (04.01.2001)

PCT

(10) International Publication Number
WO 01/01283 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US00/17989

(22) International Filing Date: 29 June 2000 (29.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/344,278 30 June 1999 (30.06.1999) US

(71) Applicant: BIZTRO, INC. [US/US]; Suite 100, 2500 Augustine Drive, Santa Clara, CA 95054 (US).

(72) Inventors: D'SOUZA, Roy, Peter; 657 Spruce Drive, Sunnyvale, CA 94086 (US). MANNING, William, Laurence; 836 Pine Hill Road, Stanford, CA 94305 (US).

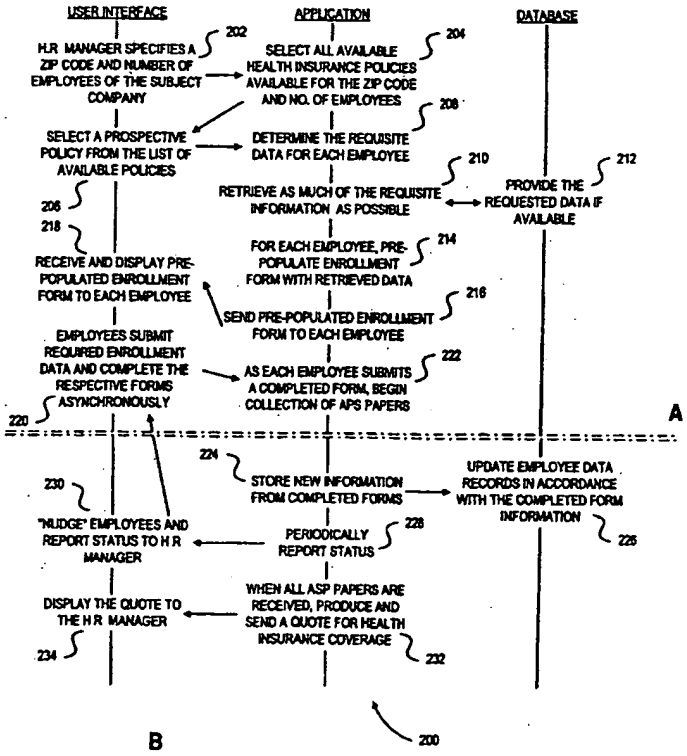
(74) Agent: IVEY, James, D.; Law Offices of James D. Ivey, 3025 Totterdell Street, Oakland, CA 94611-1742 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: IMPROVED INSURANCE QUOTE ACQUISITION MECHANISM



(57) Abstract: Requests for APS papers from physicians of employees are sent as completed insurance application forms are received asynchronously from the employees of a business concern. Accordingly, processing of APS requests is concurrent with processing of insurance application forms by employees. To expedite form completion by the employees and to reduce data entry errors by the employees, the insurance application forms are pre-populated with information about each employee already stored in an employee database maintained by the business concern. Periodic reminders are sent to those employees who delay in submitting completed forms. In addition, a human resources manager of the business concern is periodically informed regarding the status of such form submission by employees.

WO 01/01283 A2

AP

Published:

— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

IMPROVED INSURANCE QUOTE ACQUISITION MECHANISM

SPECIFICATION

CROSS REFERENCE TO RELATED APPLICATIONS

The present Application is related to the following co-pending U.S. Patent Applications, each of which is incorporated herein in its entirety by reference: (i) U.S. Patent Application S/N 09/___,___ entitled "Improved Scalable Architecture for E-Commerce Applications" by Roy D'Souza and filed June 30, 1999 (Attorney Docket No. BHUBP008); (ii) U.S. Patent Application S/N 09/___,___ entitled "Data Mining Aggregator architecture with Intelligent Selector" by Roy D'Souza and filed June 30, 1999 (Attorney Docket No. BHUBP001); (iii) U.S. Patent Application S/N 09/___,___ entitled "Data Mining With Dynamic Events" by Roy D'Souza and filed June 30, 1999 (Attorney Docket No. BHUBP002); and (iv) U.S. Patent Application S/N 09/___,___ entitled "Data Mining With Decoupled Policy From Business Application" by Roy D'Souza and filed June 30, 1999 (Attorney Docket No. BHUBP003).

FIELD OF THE INVENTION

The present invention relates to commerce systems which include computers and computer networks and, in particular, to a particularly efficient system for collecting information and receiving bids for insurance coverage.

BACKGROUND OF THE INVENTION

Currently, acquisition of insurance for companies for the benefit of employees is a tedious task. Generally, weeks or even months are required to compile information from employees and to receive an insurance premium quote using that employee information. For example, employees are typically required to provide numerous items of information and to enter this information into an application form. The employer typically must compile all such applications before forwarding the applications to the insurer. As a

result, any delay by any one employee can stall the entire insurance application process.

Once the insurer has received the applications of all employees of the applying company, the insurer sends requests to the respective physicians of the employees for Attending Physician Statements (ASPs). Receipt of all ASPs is typically a prerequisite to determining an estimated premium to be sent to the applying company as a premium quote. As a result, a delay by any one physician similarly delays the entire insurance application process.

A mechanism which reduces the time required to compile information for insurance applications and to receive a premium quote is highly desirable.

SUMMARY OF THE INVENTION

In accordance with the present invention, an integrated business e-commerce system integrates employee information from other business applications with the insurance application process for a business concern. In particular, to expedite form completion by the employees and to reduce data entry errors by the employees, the insurance application forms are pre-populated with information about each employee already stored in a employee database maintained by the business concern. Some of the data fields of the pre-populated form are read-only, i.e., are greyed to show that the employee cannot alter data represented in such data fields and alteration of the displayed data is disallowed. Accordingly, introduction of errors in data already stored in the database is prevented. In addition, integrity of such data can more easily be assured by allowing only specific people to alter specific data fields.

The overall insurance application process is further expedited by concurrently obtaining third-party information such as APS papers once one or more applications are received from the employee while other employees continue to provide information requested by the insurance application form. In particular, requests for APS papers from physicians of employees are sent as completed insurance application forms are received asynchronously from the employees of a business concern. Accordingly, processing of APS requests is concurrent with processing of insurance application forms by employees.

Specifically, a request for APS papers for each employee is sent once a completed form is received from the employee even if other employees have yet to submit completed forms.

Periodic reminders are sent to those employees who delay in submitting completed forms. In addition, a human resources manager of the business concern is periodically informed regarding the status of such form submission by employees. Accordingly, pressure can be tactfully applied to employees to encourage them to complete and submit insurance application forms.

Once all application forms and all APS papers are received, a quote for health insurance premiums for the business concern is produced and sent to the human resources manager. Since requests for APS papers are sent and processed while employee application forms are still outstanding allows the APS requests to be processed concurrently with the employee application form processing. As a result, the amount of time required to acquire the quote for health insurance premiums is significantly reduced.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system in which information required for obtaining a quote for insurance premiums is gathered in accordance with the present invention.

Figure 2 is a flow diagram illustrating the flow of information gathered in accordance with the present invention.

Figure 3 is a block diagram of components of the computer system of Figure 1 in greater detail.

Figure 4 is a block diagram of a business logic intelligent director agent of Figure 3 in greater detail.

DETAILED DESCRIPTION

In accordance with the present invention, inter-connectivity provided by a computer network is used to collect employee information and attending physician statements in an efficient manner involving substantially concurrent steps. The result is a substantial reduction in time required to receive insurance premium quotes from several

weeks to just days or even hours.

Figure 1 shows a networked computer system 100 for processing insurance applications. Computer system 100 is described more completely below. A brief description of computer system 100 is included for completeness and facilitates understanding and appreciation of the application process described herein.

A number of computers 102, 104A-B, and 124 are coupled through a wide area network 106, such as the Internet, to a scalable network server 108. In this illustrative example, computer 102 is used by a human resources manager of a business concern and the human resources manager is, in this example, interested in receiving quotes for health or life insurance premiums for employees of the business concern. The business concern of this illustrative example is sometimes referred to as the subject company. Computers 104A-B are used by respective employees of the subject company. Only two computers 104A-B used by employees are shown while it is appreciated that numerous additional computers for respective additional employees can be added. Computer 124 is used by an attending physician of one or more of the employees of the subject company and computer 124 is described more completely below.

Scalable network server 108 includes a number of individual network servers, load balancers, and routers as described more completely below. Briefly, scalable network server 108 routes data between computers 102, 104A-B, and 124 on one end and applications 110A-B and business logic 112 on the other end.

Applications 110A-B and business logic 112 access data in a database 116 through an applications programming interface (API) 114. The subject company uses applications 110A-B and remote applications which are accessible through business logic 112, such as remote application 122, to perform a number of business functions. Such business functions include, for example, payroll, accounting, benefits administration, and inter-office communications such as e-mail. Each of these functions can be performed by all or part of one or more applications such as applications 110A-B and remote application 122, each of which stores data for such functions in database 116. As a result, database 116 evolves to include an ever increasingly complete record of each employee of the subject company. While only applications 110A-B and remote application 122 are shown, it is appreciated that other applications and remote applications can be included.

Computers 102, 104A-B, and 124 access remote application 122 through business logic 112, a message queue 118, and a gateway 120. In addition, remote application 122 can access information in database 116 through gateway 120, message queue 118, and business logic 112. Briefly, business logic 112 serves requests received through wide area network 106 for information processing. Such information processing can require information processing by remote application 122. An example is described more completely below. To serve such a request, business logic 112 retrieves whatever requisite information from database 116 through API 114 and packages a request for remote application 122 and places the request on message queue 118. Message queue 118 sends the request to remote application 122 through gateway 120.

Gateway 120 can be any of a number of inter-network gateways. In one embodiment, gateway 120 includes a printer, a human operator, and a conventional communications network by which remote application 122 is reachable. For example, message queue 118 can print requests on the printer, and the human operator can send the request by facsimile or by voice communication through a telephone to an organization, e.g., an insurer, for processing. Alternatively, gateway 120 can be a wide area computer network such as the Internet.

When remote application 122 has processed the request, remote application 122 sends response information through gateway 120 which packages the response information and places it on message queue 118, addressed for business logic 112. Business logic 112 uses the response information to complete the request received through wide area network 106, storing any updated information in database 116 through API as appropriate, i.e., for any data which supersedes data already stored in database 116 or which is not already stored in database 116.

Computer system 100 has access to records of physicians 128, 130, and in computer 124. These physician records are accessible to applications 110A-B through wide area network 106 and through business logic 112, message queue 118, and gateway 120. These physician records are also accessible to remote application 122 through gateway 120; message queue 118, business logic 112, and wide area network 106; and through an additional gateway 126. Gateway 126 can be, for example, the public-switched telephone network through which a physician can be reached directly by telephone.

Flow diagram 200 (Figure 2) illustrates the process by which insurance premium quotes are acquired according to the present invention. In step 202, a human resources manager using computer 102 (Figure 1) specifies a zip code or other geographical designation of the subject company and a number of employees of the subject company. The human resources manager specifies the zip code and number of employees using, for example, hypertext markup language (HTML) forms and common gateway interface (CGI) scripts which are part of a user interface implemented by business logic 112. Business logic 112 sends the zip code and number of employees to a server application. The server application is the one of applications 110A-B and remote application 122 which processes requests for insurance premium quotes.

In step 204 (Figure 2), the server application selects all policies available to companies of having the specified zip code or other geographic designation. The server application then sends the list to business logic 112 (Figure 1).

Business logic 112 receives the list and presents the list to the human resources manager who then selects a policy from the list in step 206 (Figure 2) using the user interface of business logic 112 (Figure 1). In response to the human resources manager, business logic 112 sends data identifying the selected policy to the server application.

In step 208 (Figure 2), the server application determines the requisite information for the selected policy. Such requisite information can include information about the subject company and information about each of the employees of the subject company, including medical record information possessed by respective physicians of the employees.

In step 210, the server application retrieves as much of the requisite information as possible from database 116 (Figure 1). If the server application is one of applications 110A-B, database 116 is accessed directly through API 114. If the server application is a remote application, e.g., remote application 122, the server application accesses database 116 through gateway 120, message queue 118, and business logic 112. In response to requests by the server application, API 114 and database 116 collectively provide as much of the requested data as is available in step 212 (Figure 2).

In step 216, the server application pre-populates a form for each employee of the subject company. For example, the form can be an HTML form in which each employee uses a web browser to enter required information. In pre-populating each form, the server

application associates data retrieved from database 116 (Figure 1) with data fields required in the form so that the employee to which the form pertains does not have to fill in those particular fields. Such reduces the burden and inconvenience of the employee since the employee will frequently not have all pieces of required information on hand while much of the required information has a reasonable likelihood of being found in database 116.

The server application can also configure the form to disable alteration of specific data fields by the user so that incorrect information cannot be entered. For example, the employee can be prevented from changing data associated with data fields such as the employee's name, home address, date of birth, social security number, telephone numbers, etc. Any changes in such information should be made globally by alteration of the data stored in database 116 and under controlled circumstances, i.e., in a manner that the subject company can verify that such changes are indeed proper and properly made within database 116.

If the server application is one of applications 110A-B, the server application itself pre-populates the form and selectively enables alteration of individual data fields of the form and sends the pre-populated form through scalable network server 108 and wide area network 106 to the user. Alternatively, if the server application is a remote application such as remote application 122, business logic 112 pre-populates the form and selectively enables alteration of individual data fields of the form and sends the pre-populated form through scalable network server 108 and wide area network 106 to the user on behalf of the server application. To do so, business logic 112 receives form layout data from remote application 122, and the form layout data specifies the items of data required from individual employees. Business logic 112, or alternatively API 114, determines which data stored in database 116 is only permitted to be altered in certain, controlled circumstances, i.e., by authorized personnel responsible for the integrity of database 116. Accordingly, business logic 112 can determine which data required by the server application should be protected from alteration by the employee. If the server application is any of applications 110A-B, the server application can obtain such information directly from API 114 to properly prevent alteration of particular data by the user.

In step 218 (Figure 2), each employee of the subject company receives and views the pre-populated form received from the server application. For example, computer 104A

receives the form through wide area network 106 and scalable network server 108 and displays the form for the employee who uses computer 104A. The form can be displayed, for example, using a conventional web browser such as the Netscape Communicator available from Netscape Communications Corporation of Mountain View, California or the Internet Explorer available from Microsoft Corporation of Redmond, Washington. While web browsers implement a pull paradigm in which information is generally presented only when requested, it is preferred that some push-paradigm communications be used to bring the pre-populated form to the employee's attention without requiring a request for the form by the employee. Such push-paradigm communications can include, for example, electronic messaging such as e-mail to send a reminder to request and view the form or, alternatively, redirection from, or a server-side include within, a frequently viewed document to thereby cause the document to automatically display the pre-populated form for the employee. Such redirection and server-side includes are widely used and well known features of HTML.

In step 220 (Figure 2), employees of the subject independently supply data requested by the pre-populated form and submit completed forms. The server application receives the forms asynchronously over a period of time as individual employees submit each respective completed form.

In step 222 (Figure 2), the server application begins collection of attending physician statement (APS) papers as each completed form is received. Specifically, the server application initiates collection of APS papers for one employee who has submitted a completed form even before a completed form is subsequently received from another employee. Thus, collection of APS papers and filling in of pre-populated forms by employees are concurrent.

The server application can begin collection of APS papers in a number of ways. For example, if the server application is any of applications 110A-B, the server application can send a request for APS papers to a physician at computer 124 through scalable network server 108 and wide area network 106 or, alternatively, through API 114, business logic 112, message queue 118, and gateway 120 to a physician 130. Similarly, if the server application is a remote application such as remote application 122, the server application can send a request for APS papers (i) through gateway 120, message queue

118, business logic 112, scalable network server 108, and wide area network 106 to a physician at computer 124; (ii) through gateway 120 to physician 130; and (iii) through a separate gateway 126, which is analogous to gateway 120, to a physician 128.

Each physician, in response to such requests, begins collection of APS papers for the employees of the subject company. In addition, since such requests are issued as completed forms are received asynchronously, some physicians can begin preparation of APS papers while some employees of the subject company have yet to complete the pre-populated forms.

The server application also updates information stored in database 116 for those employees who have submitted completed forms in step 224 (Figure 2). In particular, any items of information which are included in the completed form which are not already stored in database 116 (Figure 1) are written to database 116. If the server application is any of applications 110A-B, the server application stores such data directly through API 114 which in turn directly accesses database 116. If the server application is a remote application such as remote application 122, business logic 112 reviews the items of information of the completed form, identifies those items of information which were not retrieved from database 116 to pre-populate the form, and stores those items in database 116 through API 114. In step 226 (Figure 2), API 114 and database 116 cooperate to store the data in database 116.

While physicians continue to prepare APS papers and employees continue to fill out and submit the pre-populated forms, the server application periodically determines the status of the forms submission by employees and the APS paper gathering by physicians. The period of such status determination can be, for example, once, twice, or thrice each week. If the server application is a remote application such as remote application 122, business logic 112 maintains a list of employees who have submitted completed forms in the manner described above and can therefore determine which employees of the subject company have yet to submit completed forms. Business logic 112 also receives periodic reports from remote application 122 as to which of the physicians have submitted APS papers as required. If the server application is any of applications 110A-B, the server application maintains a list of which employees have submitted completed forms and which physicians have submitted proper APS papers.

At each period, the server application reports status to business logic 112 in step 224 (Figure 2). In step 226, business logic 112 (Figure 1) reports the same status information to the human resources manager, e.g., as an e-mail message to computer 102. In addition, business logic 112 sends reminder messages, e.g., e-mail messages, to those employees who have yet to submit a completed form. Thus, while physicians are compiling APS papers for those employees who have submitted completed forms, other employees are encouraged to complete and submit their respective pre-populated forms.

As a result of such timely reminders, all employees eventually submit completed forms. Such is assured by the periodic status reports to the human resources manager. The amount of time required to collect completed forms from all employees and to collect all APS papers from all physicians is reduced from weeks or months using conventional techniques to days or even hours using the technique represented in Figure 2. When all completed forms and all APS papers are received, the server application produces a quote for health insurance coverage for the employees of the subject company and sends the quote to the human resources manager at computer 102. The manner in which the quote is produced from the completed forms and APS papers is conventional. When the quote is received by computer 102, the quote is displayed for the human resources manager.

Thus, according to flow diagram 200, the amount of time required to get a quote for insurance is dramatically reduced.

Scalable Architecture

The architecture of scalable network server 108, applications 110A-B, business logic 112, API 114, and database 116 is shown in greater detail in Figure 3.

In particular, Figure 3 shows a clustered computer system architecture wherein an intelligent director agent (IDA) is included with each of the clusters that implement a webserver stage, a business logic stage, and a data repository stage. Preferably, there is an IDA for each cluster, although more than one cluster may be provided per stage, in which case multiple IDAs may be provided. Furthermore, as will be discussed later herein, the clusters may be disposed at one local site or may be dispersed among geographically remote locations. Note that although Figure 3 shows an intelligent director agent for each of these stages, it is contemplated that in some clustered computer systems, not every stage

needs to be provided with an intelligent director agent and that significant benefits may be achieved by endowing even only one of the stages with one or more intelligent director agents. Conversely, a stage may comprise multiple clusters, in which case multiple IDAs may be provided.

With reference to Figure 3, there is shown portions of computer system 100, which is typically connected to wide area network 106 as described above. In particular, Figure 3 shows a webserver stage 304 which includes scalable server 108, a business logic stage 306 which includes business logic 112 and applications 110A-B, and a data repository stage 308 which includes API 114 and database 116.

Data repository stage 308 represents the stage wherein data for use by the business logic software modules are kept and includes the data stores as well as the database logic employed to access the data stores. Business logic stage 306 represents the stage wherein the computer cluster(s) employed to execute business logic 112 and applications 110A-B is implemented. For simplicity, only one cluster comprising four business logic servers is shown in Fig. 3. Webserver stage 304 represents the stage wherein the computer cluster(s) employed to execute scalable network server 108 is implemented. Webserver stage 304 generally facilitates the users' interaction with the rest of computer system 100 using the web-based paradigm or a suitable paradigm for interacting with wide area network 106 (Figure 1). Again, only one cluster comprising five webservers is shown in Fig. 3 to simplify the illustration.

In the case of Fig. 3, the servers within each stage and within each cluster may be heterogeneous (i.e., implemented on different platforms and having different capability) and each may operate a different set of business logic modules, i.e., application software modules. By way of example, business logic 112 and applications 110A-C within business logic stage 306 may be implemented using different hardware/software platforms and configurations that are adapted for operating business logic 112 and applications 110A-B implemented therein. In other words, there is no requirement in the present invention that the servers associated with a given stage or cluster or even those running copies of a particular software module be homogeneous (although such can be readily accommodated by the instant computer system architecture without any major modification). As long as the servers in a cluster can communicate with the IDA that is

associated with that cluster and can be adapted to operate cooperatively with one another within a cluster, the servers can be implemented in the cluster architecture of the present invention. It should be noted that the technologies, protocols, and methodologies exist for allowing heterogeneous computers to communicate and work cooperatively and will not be discussed in greater detail herein.

Beginning with the user's access request via path 310 (by, for example, typing in the Uniform Resource Locator or URL at the user's web browser in computer 102A — Figure 1), the request is forwarded to a webserver logic intelligent director agent (IDA) 312, which decides among the webservers 314a-314e as to which of these webservers should service this user's access request. As a threshold determination, webserver logic IDA 312 may ascertain whether the user had recently accessed the service through a particular webserver of webserver stage 304. If so, there may be data pertaining to this user that is cached at that particular webserver, and it may be more efficient to continue assigning this user to that webserver to take advantage of the cached data.

On the other hand, if it is determined that this user has not recently accessed the service or if there is no cached data pertaining to this user on any of the webservers, webserver logic IDA 312 may assign the user to one of webservers 314a-314e. The decision of which webserver to assign may be made based on the current relative load levels on the respective webservers, the information pertaining to which is periodically received by webserver logic IDA 312 from the webservers through path 332. Additionally, webserver logic IDA 312 also receives additional information pertaining to the webservers and the webserver logic software modules implemented on the webservers to facilitate improved access speed and reliability. Thus, the webserver logic IDA 312 arbitrates among the webserver computers based not only on the relative load level information associated with the individual webservers but also based on information pertaining to the individual webserver logic software modules.

The assigned webserver may authenticate the user to ascertain whether the user is registered and/or properly authorized to use the service offered through computer system 100. After successful authentication, if the user subsequently indicates that he wishes to employ a particular business logic software. i.e., a particular one of applications 110A-B (by, for example, inputting data or taking an action that requires the attention of a

particular business logic module), the webserver assigned to him then accesses a business logic IDA 340 to ascertain the appropriate business logic server (i.e., the appropriate server in the business logic stage such as one of business logic 112 or applications 110A-C) to which the user's transaction request may be sent.

The decision pertaining to which business logic server to assign may be made based on the current relative load levels on the respective business logic servers, the information pertaining to which is periodically received by business logic IDA 340 from the business logic servers through path 342. Additionally, business logic IDA 340 also receives additional information pertaining to the business logic servers and more importantly the business logic software modules implemented on the business logic servers to facilitate improved access speed and reliability. Accordingly, the routing decision taken by the business logic; IDA is based not only on information pertaining to the individual business logic servers but also based on information pertaining to the individual business logic software modules implemented thereon.

The availability of the additional business logic server-specific information and the business logic module-specific information also facilitates inventive techniques to improve access speed and reliability during software upgrades, to maintain a desired level of fault tolerance for the business logic software and/or the business logic servers, to reactively and/or prospective load balance among the business logic servers, and to efficiently employ remote business logic servers to accomplish improving access speed and reliability. Some of the additional data kept by the business logic IDA and their roles in improving access speed and reliability in accordance with embodiments of the present invention will be discussed later herein.

Each of the business logic software programs, i.e., business logic 112 and applications 110A-B, has many copies distributed among the servers of the cluster to facilitate redundancy and scalability.

Once a business logic server having thereon the requisite business logic module to service the user's transaction request is assigned to service the incoming transaction request, subsequent traffic between the webserver assigned earlier to that user and the assigned business logic server may be (but is not required to be) transmitted directly without going through the assigned business logic IDA.

If the business logic module employed by the user requires data from data repository stage 308, the business logic software module, through the business logic server, may consult yet another IDA (shown in Fig. 3 as database logic IDA 350), which picks the most suitable database server 352, 354, and/or 356 for serving up the data. The decision regarding which database server to assign may be made based on the current relative load level on the respective database servers that have the necessary data, the information pertaining to which is periodically received by database logic intelligent director agent 350 from the database servers through path 360. Like the business logic IDA and the webserver IDA, however, the database logic IDA 350 also receives additional information pertaining to the database servers as well as the database server logic modules implemented on the database servers to facilitate improved access speed and reliability. Once a database server having thereon the requisite data to service the user's transaction request is assigned, subsequent traffic between the business logic server that requests the data and the assigned database server may be (but is not required to be) transmitted directly without going through the assigning database logic IDA.

In one embodiment, an IDA may be co-located with the router that routes the traffic to the servers of the cluster, or it may be implemented separately from the router. It should be kept in mind that although Fig. 3 shows an IDA for each of the webserver stage, the business logic stage, and the data repository state, there is no requirement that there must be an IDA for each stage, or each cluster for that matter if there are multiple clusters per stage. The provision of an IDA, even with only one cluster or one stage of the clustered computer system, dramatically improves access speed and reliability even when other clusters and stages may be implemented without IDAs.

As mentioned earlier, an intelligent directory agent (IDA) receives more than just load status data from the servers it services. With reference to business logic intelligent director agent (IDA) 340, for example, it is preferable that the business logic IDA tracks one or more of the additional information such as server processing capability, server geographic identification (e.g., remote or local to the site that implements the webserver stage and/or the data repository stage), the average latency for servicing a transaction request (e.g., due to the server's geographic remoteness or the speed of the network connection), the list of business logic modules that are compatible with each server, the

list of the business logic modules actually implemented on each server, the version of the business logic modules implemented, and/or the load experienced by the business logic modules on the servers. In one embodiment, the business logic IDA also receives information pertaining to external historical profiles (368) of transaction requests and processing loads on the business logic modules and/or the business logic servers in order to predict usage demands placed on the business logic modules and to prospectively balance the loads among the business logic servers if needed so that an anticipated surge in usage does not overwhelm any particular business logic module.

Fig. 4 illustrates, in accordance with one embodiment of the present invention, a simplified logic block diagram of an exemplary business logic intelligent director agent (IDA) 340. Although only the business logic IDA is described in details herein, the webserver logic IDA and the database logic IDA may be similarly formed. However, their similar construction will not be discussed in details for brevity sake. With reference to Fig. 4, business logic requests from the webservers are received by business logic IDA 340 via path 370. Within business logic intelligent director agent 340, both server-specific and software-specific information is received and maintained in addition to the relative load status on individual business logic servers.

Some of the additional pieces of information are received from the business logic servers via path 342 and stored in exemplary blocks 404, 406, 408, 410, 412, 414, and 416, respectively. For ease of illustration, not every piece of information is shown in Fig. 4. Note that some of information is static and may be received as part of the registration process that the servers underwent as they were installed into the cluster. Examples of such static information includes server processing capability and business logic module version number. Other information may be dynamically received by the IDA from the servers (such as the list of business logic modules implemented on each server) and other network monitoring tools (such as conventional software tools that track network congestion at specific locations). Still, other information may be derived from the information received dynamically and/or statically (such as the average latency time for servers, which may be calculated periodically based on average network latency between the webserver and the business logic server, the average network latency between the business logic server and the available database cluster, the processing capability of the

servers, and the like).

Business server directory 404 may track information pertaining to the list of business logic servers available to the clustered computer system, their remote/local status, their certified/uncertified status (which may be expressed as Boolean values or may be a numerical value that reflects their preference in receiving and servicing transaction requests), the list of business logic servers capable of being loaded with a particular business logic software, the list of business logic servers capable of being used for running a particular business logic module, their relative weight which reflects the relative preference with which traffic should be directed to the individual servers (e.g., due to network conditions or other factors), and the like.

Business logic module version block 406 may track information pertaining to the software versions of the business logic modules implemented on the various business logic servers. Further, business logic version block 406 may track information pertaining to the certified/uncertified status of each copy of the business logic modules, the relative weight of each copy of business logic module which reflects the relative preference with which traffic should be directed to it, and the like.

Business logic module load status block 408 may track information pertaining to the level of load currently experienced by the individual business logic modules (in number of transactions per second or the number of users currently using a business logic module, for example). This information may be tracked for business logic modules currently in operation, individually and/or as a group average.

Server processing capacity block 410 may track the processing capability (again in number of transactions per second or the number users that can be supported concurrently) of the individual business logic servers in order to ascertain how much bandwidth a particular server may have, how much has been used, and how much is available.

Business logic server load status block 412 may track a similar type of data as business logic module load status, albeit at the server level instead of the business logic module level. Business logic server average latency block 414 may track the average latency to be expected if a particular business logic server is employed to service the transaction request. The average latency may be calculated based on the processing capability of the server, how remote it is from the webserver that issues the transaction

request (which is impacted by network latency), how remote it is from the database that may be needed to service the transaction request (which is also impacted by network latency). Business logic server log file block 416 may track the operational status of the business logic server and/or the business logic modules implemented thereon to determine, for example, the length of time that the server and/or the business logic module has been in operation without failure and other types of log file data.

Business logic intelligent director agent 340 also includes a data mining module 430, which receives the external historical profiles (368 of Fig. 3) of past usage trends on the various business logic modules and/or business logic servers, and ascertains prospectively the load demand on the various business logic modules and/or business logic servers. Data mining module 430 may be implemented using a variety of available data mining methodologies.

Using the server-specific and the business logic module-specific information available, a business logic selector module 434 then selects one of the business logic servers to service the pending business logic request and transmits the selection to the requesting webserver via path 372.

Within business logic intelligent director agent 340, there is also shown a configuration module 440, representing the module that either reactively or prospectively reconfigures and/or reshuffles the business logic modules among the business logic servers to permit the clustered computer system to better handle the processing load and to achieve the desired level of fault tolerance.

The above description is illustrative only and is not limiting. The present invention is limited only by the claims which follow.

****

What is claimed is:

1. A method for acquiring a quote for insurance premiums for a business concern which includes two or more employees, the method comprising:

determining that two or more required items of information are required of each of the employees;

retrieving one of more of the required items of information from a database of employee information;

including, in a respective request for information for each employee, identification of one or more other ones of the required items to be supplied by the employees; and

submitting the respective requests for information to each of the employees.

2. The method of Claim 1 wherein information from a third party other than the business concern and a provider of the quote is required, the method further comprising:

receiving a first information request response from a first of the employees;

subsequently receiving a subsequent information request response from a second of the employees; and

sending a request to the third party for information pertaining to the first employee prior to receiving the subsequent information request response.

3. The method of Claim 2 wherein the third party is an attending physician of one or more of the employees; and

further wherein the information pertaining to the first employee is an attending physician statement.

4. The method of Claim 2 further comprising:

sending a request for a second attending physician statement for the second employee.

5. The method of Claim 4 further comprising:

receiving attending physician statements for each of the employees; and

generating a quote for health insurance premiums from the information request responses and the received attending physician statements.

6. The method of Claim 2 further comprising:

determining that the first information request response includes new information pertaining to the first employee which is not represented in the database; and

storing the new information in the database.

7. The method of Claim 1 further comprising:

sending periodic reminders to those of the employees for which an information request response has not been received.

8. The method of Claim 1 further comprising:

sending periodic status reports to the business concern regarding the status of each of the employees with respect to receipt from such employee of an information request response.

9. A computer readable medium useful in association with a computer which includes a processor and a memory, the computer readable medium including computer instructions which are configured to cause the computer to acquire a quote for insurance premiums for a business concern which includes two or more employees by:

determining that two or more required items of information are required of each of the employees;

retrieving one of more of the required items of information from a database of employee information;

including, in a respective request for information for each employee, identification of one or more other ones of the required items to be supplied by the employees; and

submitting the respective requests for information to each of the employees.

10. The computer readable medium of Claim 9 wherein information from a third party other than the business concern and a provider of the quote is required; and

wherein the computer instructions are further configured to cause the computer to acquire a quote for insurance premiums by:

receiving a first information request response from a first of the employees;

subsequently receiving a subsequent information request response from a second of the employees; and

sending a request to the third party for information pertaining to the first employee prior to receiving the subsequent information request response.

11. The computer readable medium of Claim 10 wherein the third party is an attending physician of one or more of the employees; and

further wherein the information pertaining to the first employee is an attending physician statement.

12. The computer readable medium of Claim 10 wherein the computer instructions are further configured to cause the computer to acquire a quote for insurance premiums by:

sending a request for a second attending physician statement for the second employee.

13. The computer readable medium of Claim 12 wherein the computer instructions are further configured to cause the computer to acquire a quote for insurance premiums by:

receiving attending physician statements for each of the employees; and

generating a quote for health insurance premiums from the information request responses and the received attending physician statements.

14. The computer readable medium of Claim 10 wherein the computer instructions are further configured to cause the computer to acquire a quote for insurance premiums by:

determining that the first information request response includes new information pertaining to the first employee which is not represented in the database; and

storing the new information in the database.

15. The computer readable medium of Claim 9 wherein the computer instructions are further configured to cause the computer to acquire a quote for insurance premiums by:

sending periodic reminders to those of the employees for which an information request response has not been received.

16. The computer readable medium of Claim 9 wherein the computer instructions are further configured to cause the computer to acquire a quote for insurance premiums by:

sending periodic status reports to the business concern regarding the status of each of the employees with respect to receipt from such employee of an information request response.

17. A computer system comprising:

a processor;

a memory operatively coupled to the processor; and

an insurance quote acquisition module (i) which executes in the processor from the memory and (ii) which, when executed by the processor, causes the computer to acquire a quote for insurance premiums for a business concern which includes two or more employees by:

determining that two or more required items of information are required of each of the employees;

retrieving one of more of the required items of information from a

database of employee information;

including, in a respective request for information for each employee, identification of one or more other ones of the required items to be supplied by the employees; and

submitting the respective requests for information to each of the employees.

18. The computer system of Claim 17 wherein information from a third party other than the business concern and a provider of the quote is required; and

wherein the insurance quote acquisition module is further configured to cause the computer to acquire a quote for insurance premiums by:

receiving a first information request response from a first of the employees;

subsequently receiving a subsequent information request response from a second of the employees; and

sending a request to the third party for information pertaining to the first employee prior to receiving the subsequent information request response.

19. The computer system of Claim 18 wherein the third party is an attending physician of one or more of the employees; and

further wherein the information pertaining to the first employee is an attending physician statement.

20. The computer system of Claim 18 wherein the insurance quote acquisition module is further configured to cause the computer to acquire a quote for insurance premiums by:

sending a request for a second attending physician statement for the second employee.

21. The computer system of Claim 20 wherein the insurance quote acquisition

module is further configured to cause the computer to acquire a quote for insurance premiums by:

receiving attending physician statements for each of the employees; and

generating a quote for health insurance premiums from the information request responses and the received attending physician statements.

22. The computer system of Claim 18 wherein the insurance quote acquisition module is further configured to cause the computer to acquire a quote for insurance premiums by:

determining that the first information request response includes new information pertaining to the first employee which is not represented in the database; and

storing the new information in the database.

23. The computer system of Claim 17 wherein the insurance quote acquisition module is further configured to cause the computer to acquire a quote for insurance premiums by:

sending periodic reminders to those of the employees for which an information request response has not been received.

24. The computer system of Claim 17 wherein the insurance quote acquisition module is further configured to cause the computer to acquire a quote for insurance premiums by:

sending periodic status reports to the business concern regarding the status of each of the employees with respect to receipt from such employee of an information request response.

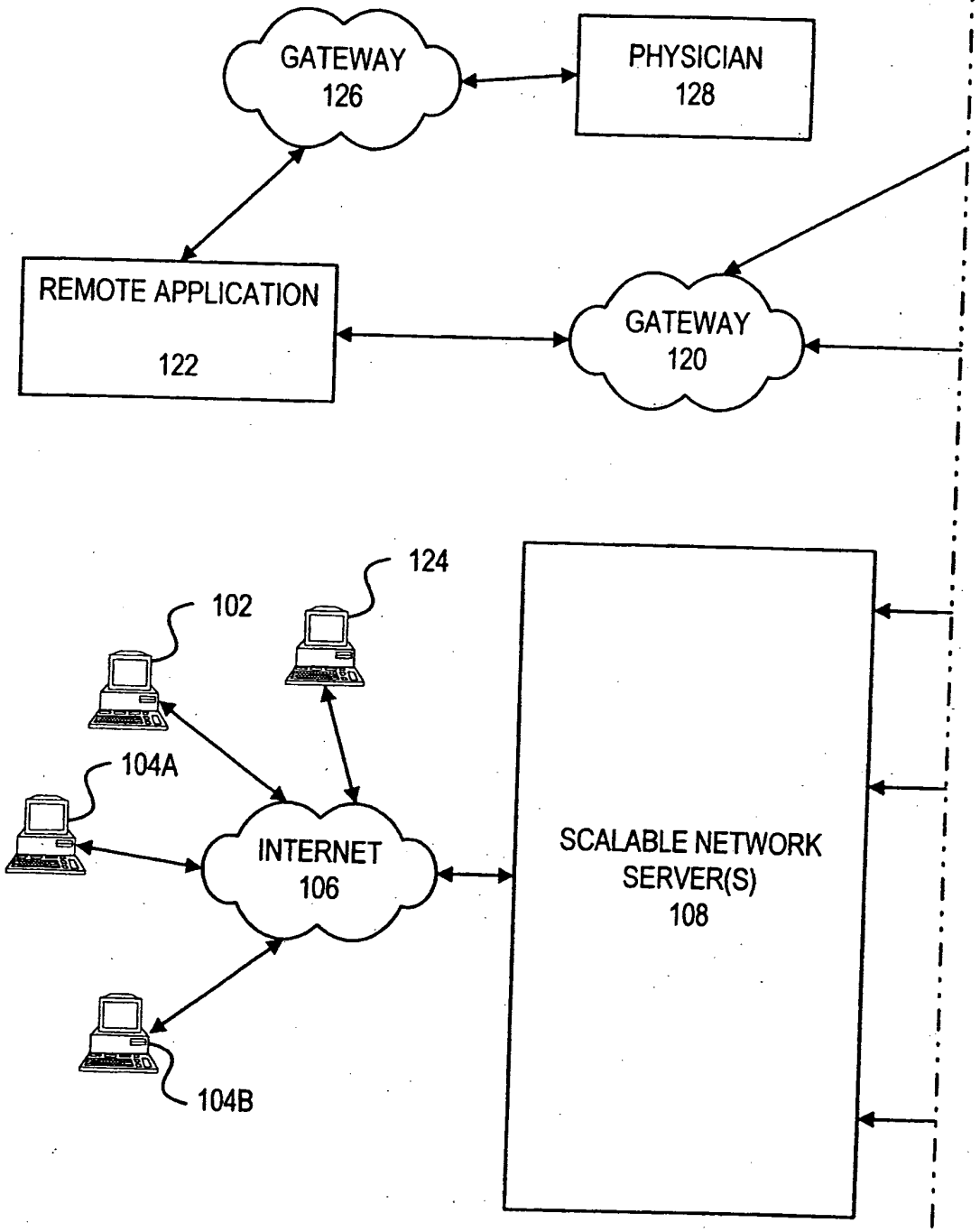

GATEWAY
126
PHYSICIAN
128
REMOTE APPLICATION
122
GATEWAY
120
124
102
104A
INTERNET
106
SCALABLE NETWORK
SERVER(S)
108
104B

FIGURE 1A

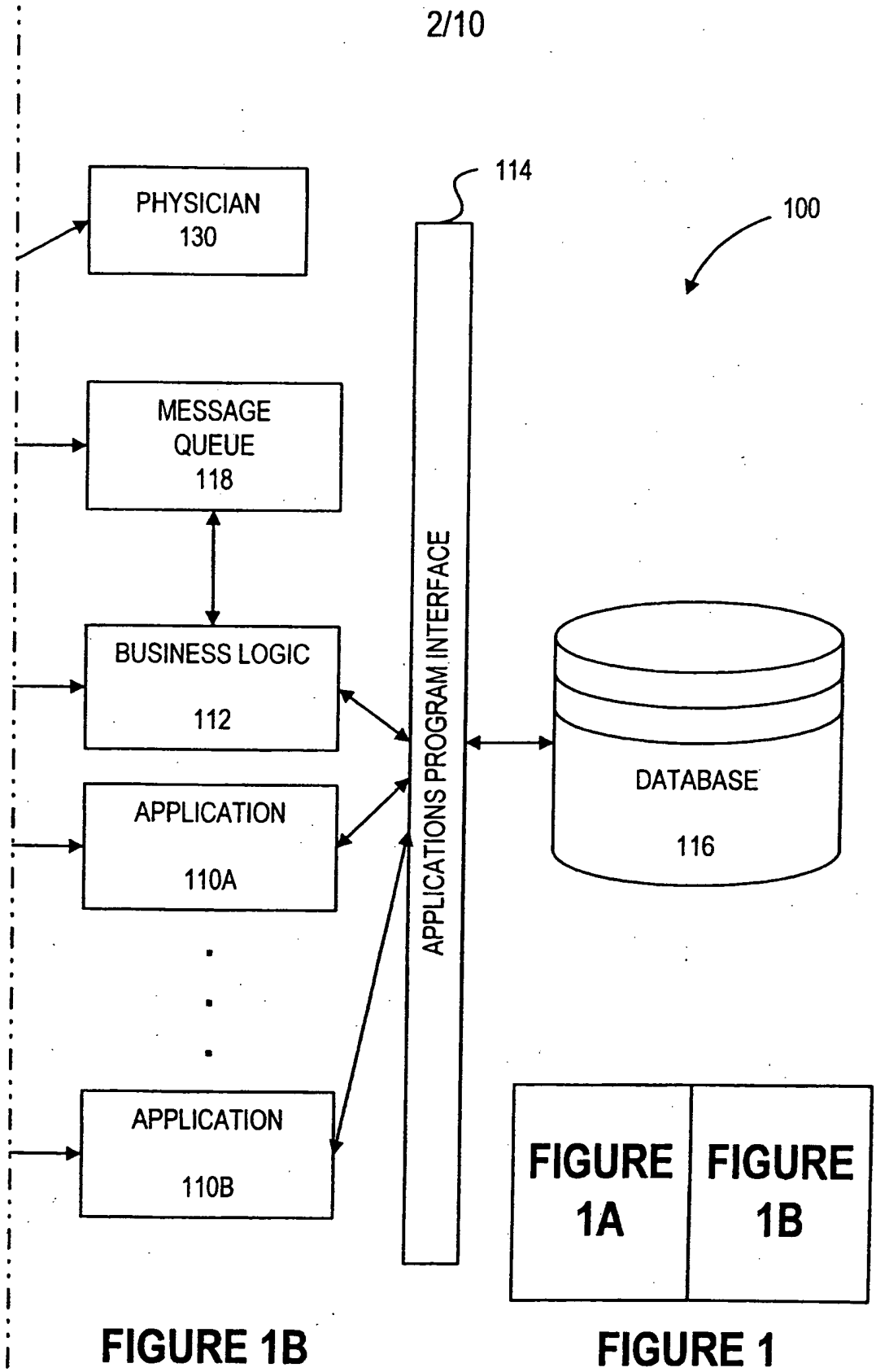


FIGURE 1B

FIGURE 1A	FIGURE 1B

FIGURE 1

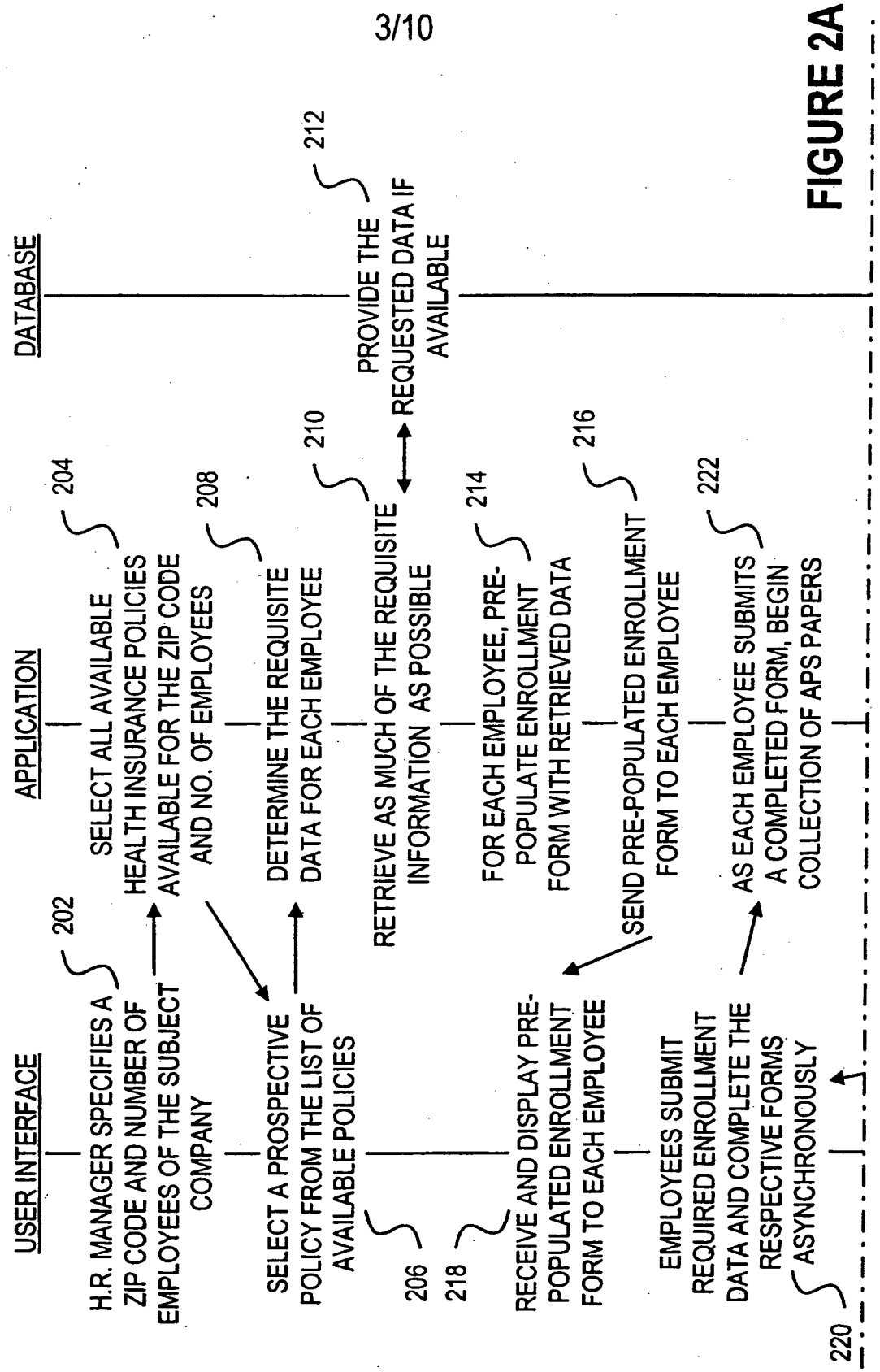

USER INTERFACE
APPLICATION
DATABASE
H.R. MANAGER SPECIFIES A ZIP CODE AND NUMBER OF EMPLOYEES OF THE SUBJECT COMPANY
202
SELECT ALL AVAILABLE HEALTH INSURANCE POLICIES AVAILABLE FOR THE ZIP CODE AND NO. OF EMPLOYEES
204
DETERMINE THE REQUISITE DATA FOR EACH EMPLOYEE
208
RETRIEVE AS MUCH OF THE REQUISITE INFORMATION AS POSSIBLE
210
PROVIDE THE REQUESTED DATA IF AVAILABLE
212
SELECT A PROSPECTIVE POLICY FROM THE LIST OF AVAILABLE POLICIES
206
FOR EACH EMPLOYEE, PRE-POPULATE ENROLLMENT FORM WITH RETRIEVED DATA
214
RECEIVE AND DISPLAY PRE-POPULATED ENROLLMENT FORM TO EACH EMPLOYEE
218
SEND PRE-POPULATED ENROLLMENT FORM TO EACH EMPLOYEE
216
EMPLOYEES SUBMIT REQUIRED ENROLLMENT DATA AND COMPLETE THE RESPECTIVE FORMS ASYNCHRONOUSLY
220
AS EACH EMPLOYEE SUBMITS A COMPLETED FORM, BEGIN COLLECTION OF APS PAPERS
222

FIGURE 2A

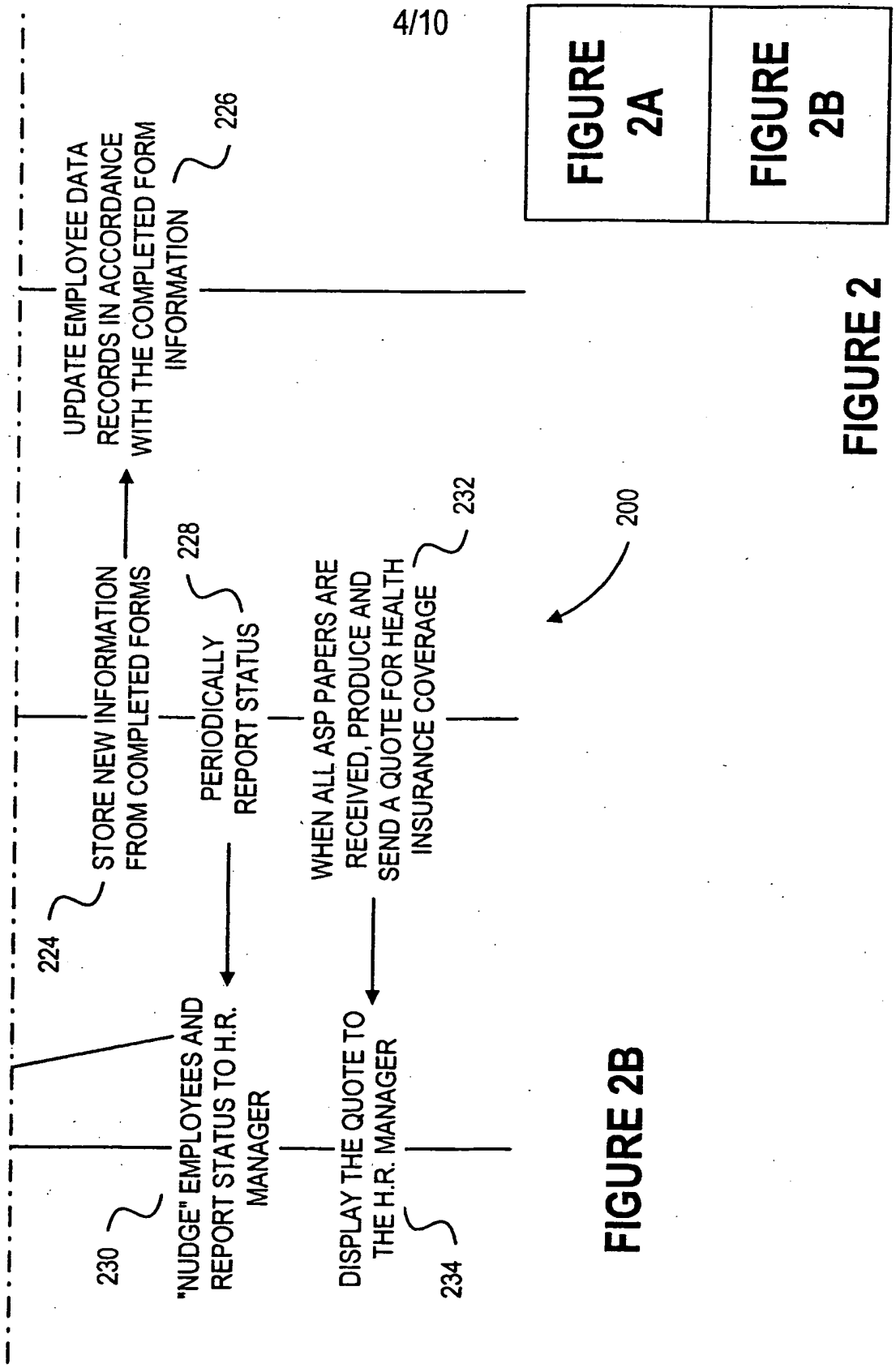


FIGURE 2B

FIGURE 2

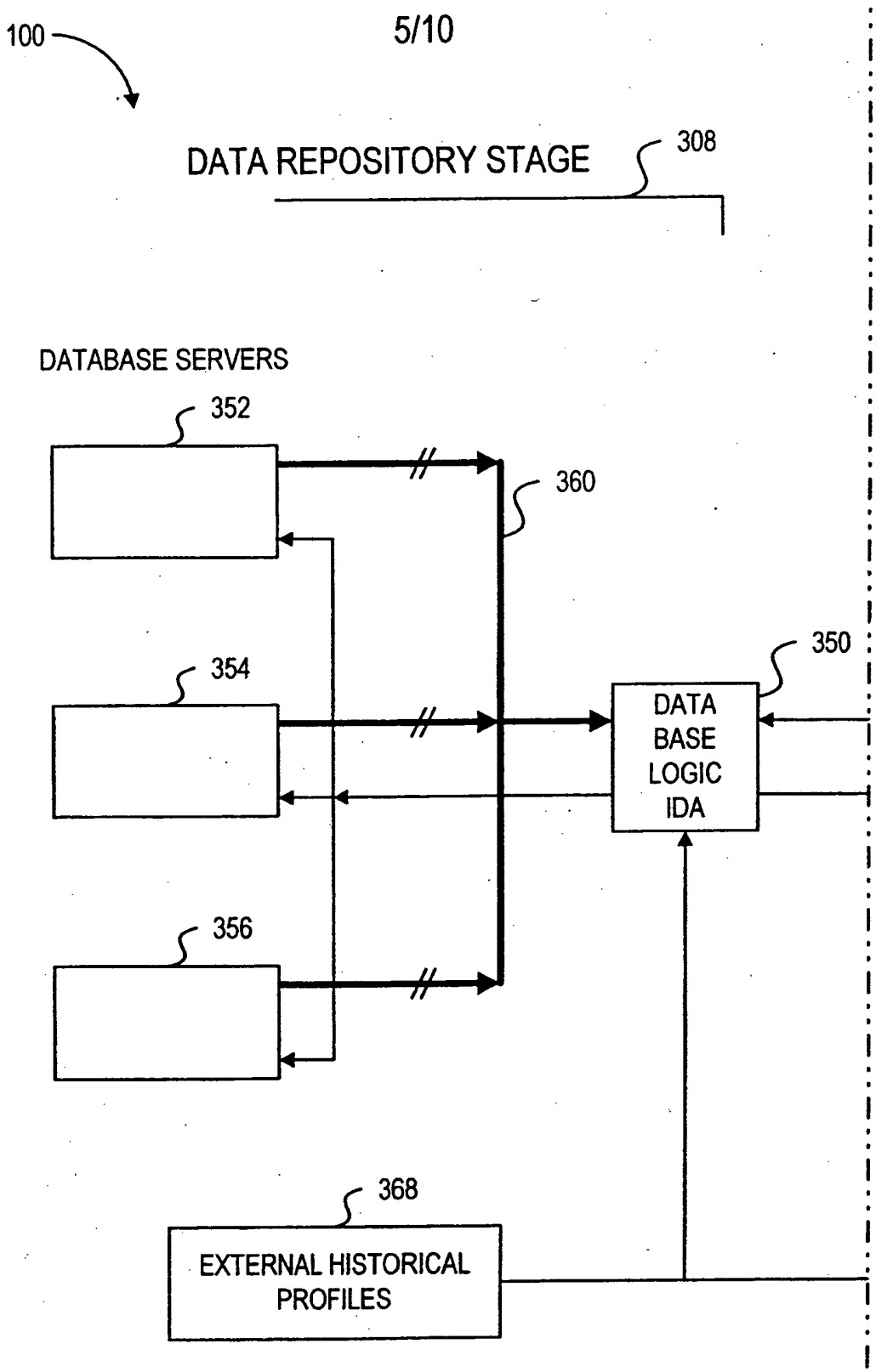

100
DATA REPOSITORY STAGE
308
DATABASE SERVERS
352
360
354
350
DATA BASE LOGIC IDA
356
368
EXTERNAL HISTORICAL PROFILES

FIGURE 3A

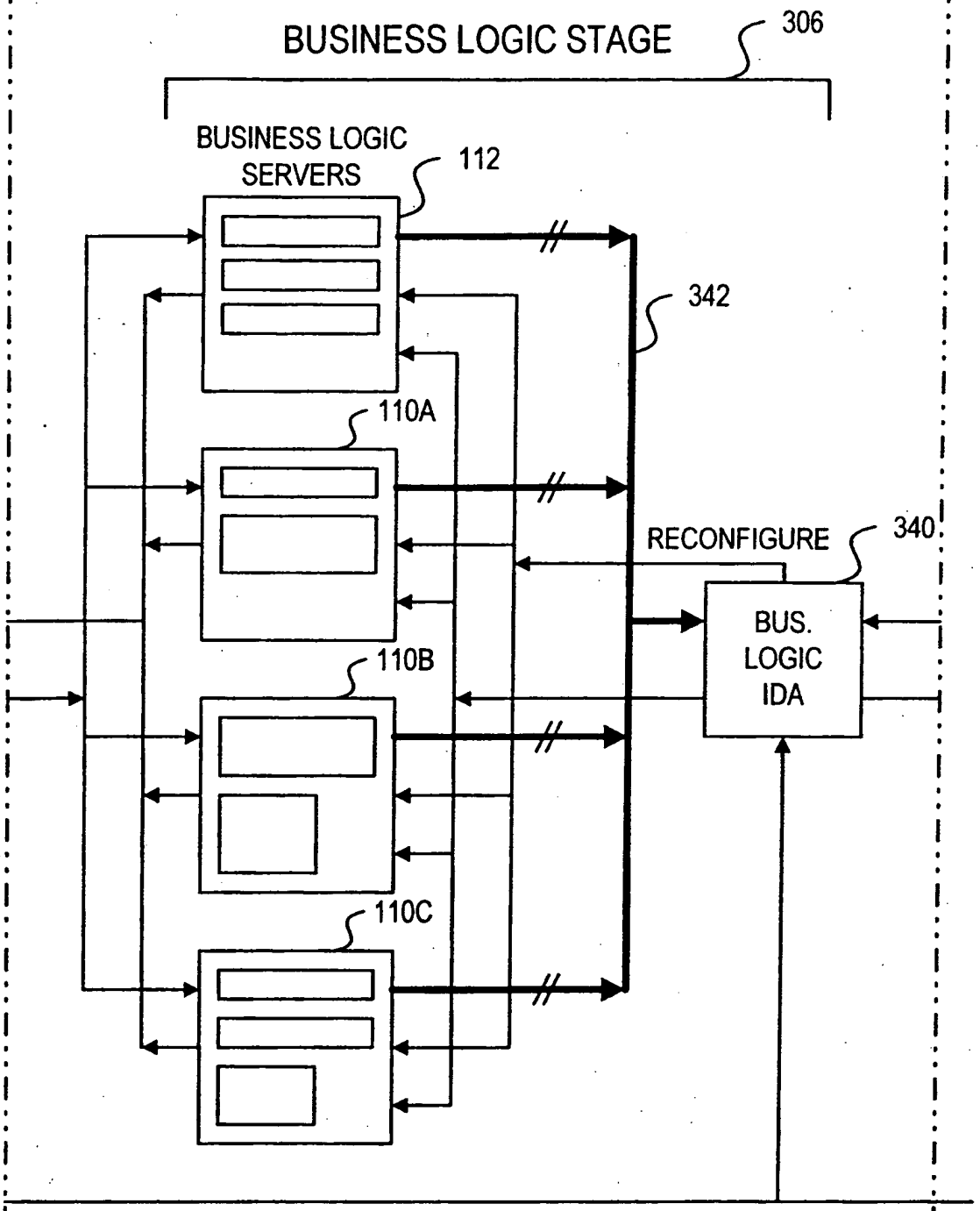

BUSINESS LOGIC STAGE
306
BUSINESS LOGIC SERVERS
112
342
110A
110B
110C
RECONFIGURE
340
BUS. LOGIC IDA

FIGURE 3B

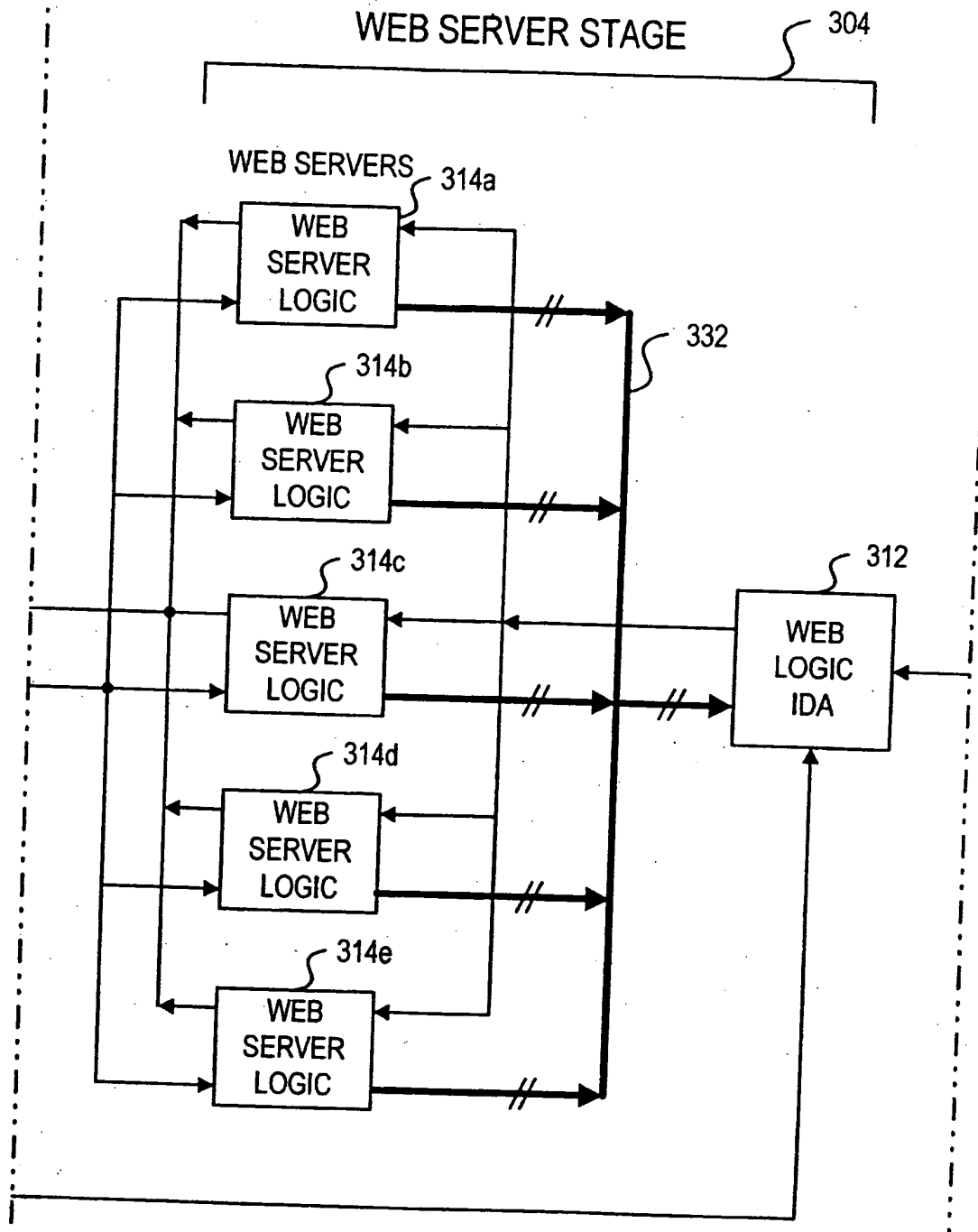

WEB SERVER STAGE
304
WEB SERVERS
314a
WEB SERVER LOGIC
332
314b
WEB SERVER LOGIC
314c
WEB SERVER LOGIC
312
WEB LOGIC IDA
314d
WEB SERVER LOGIC
314e
WEB SERVER LOGIC

FIGURE 3C

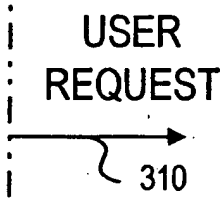


FIGURE 3D

FIGURE 3A	FIGURE 3B	FIGURE 3C	FIGURE 3D

FIGURE 3

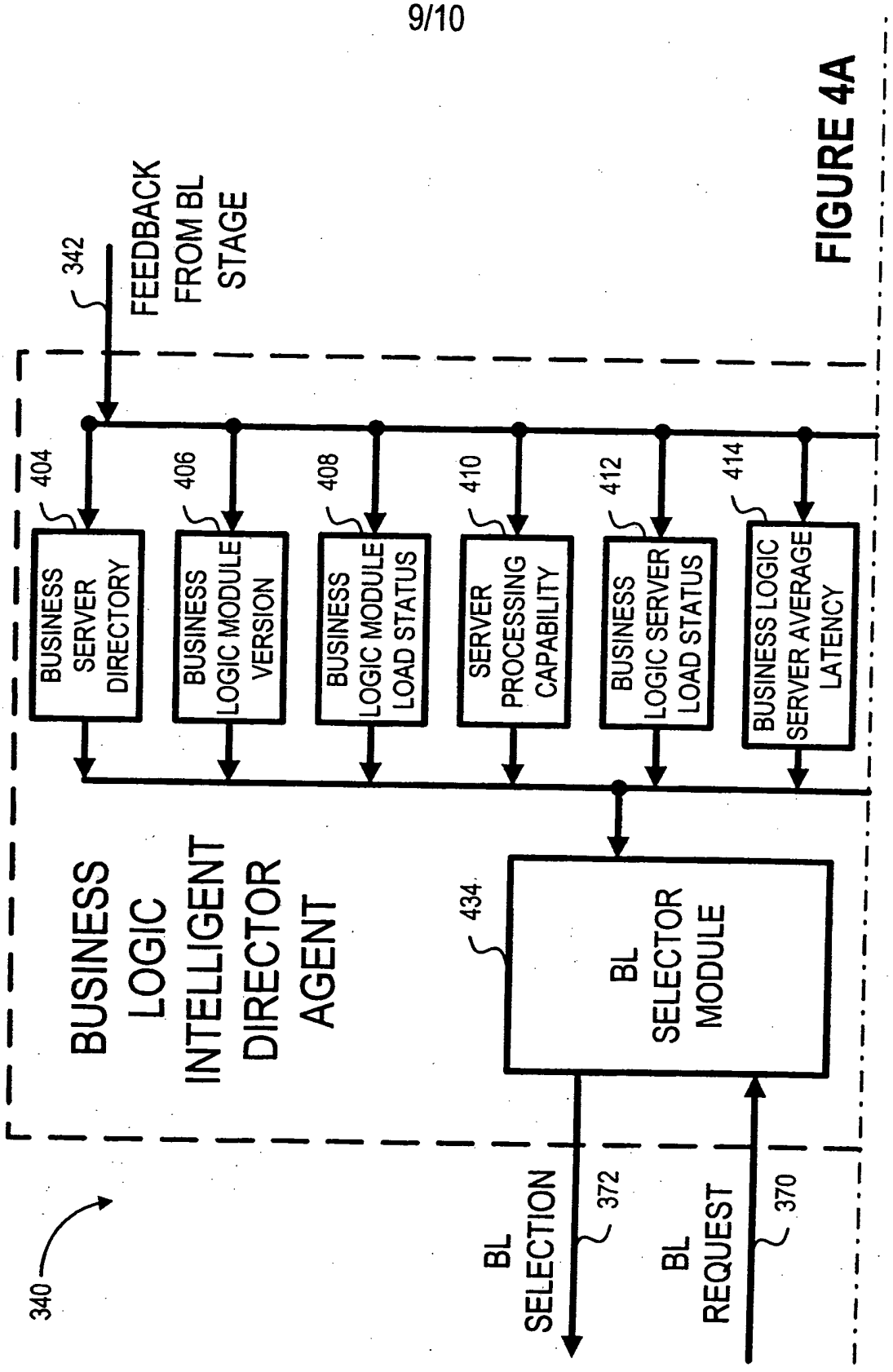

340
BUSINESS LOGIC INTELLIGENT DIRECTOR AGENT
342
FEEDBACK FROM BL STAGE
404
BUSINESS SERVER DIRECTORY
406
BUSINESS LOGIC MODULE VERSION
408
BUSINESS LOGIC MODULE LOAD STATUS
410
SERVER PROCESSING CAPABILITY
412
BUSINESS LOGIC SERVER LOAD STATUS
414
BUSINESS LOGIC SERVER AVERAGE LATENCY
434
BL SELECTOR MODULE
372
BL SELECTION
370
BL REQUEST

FIGURE 4A

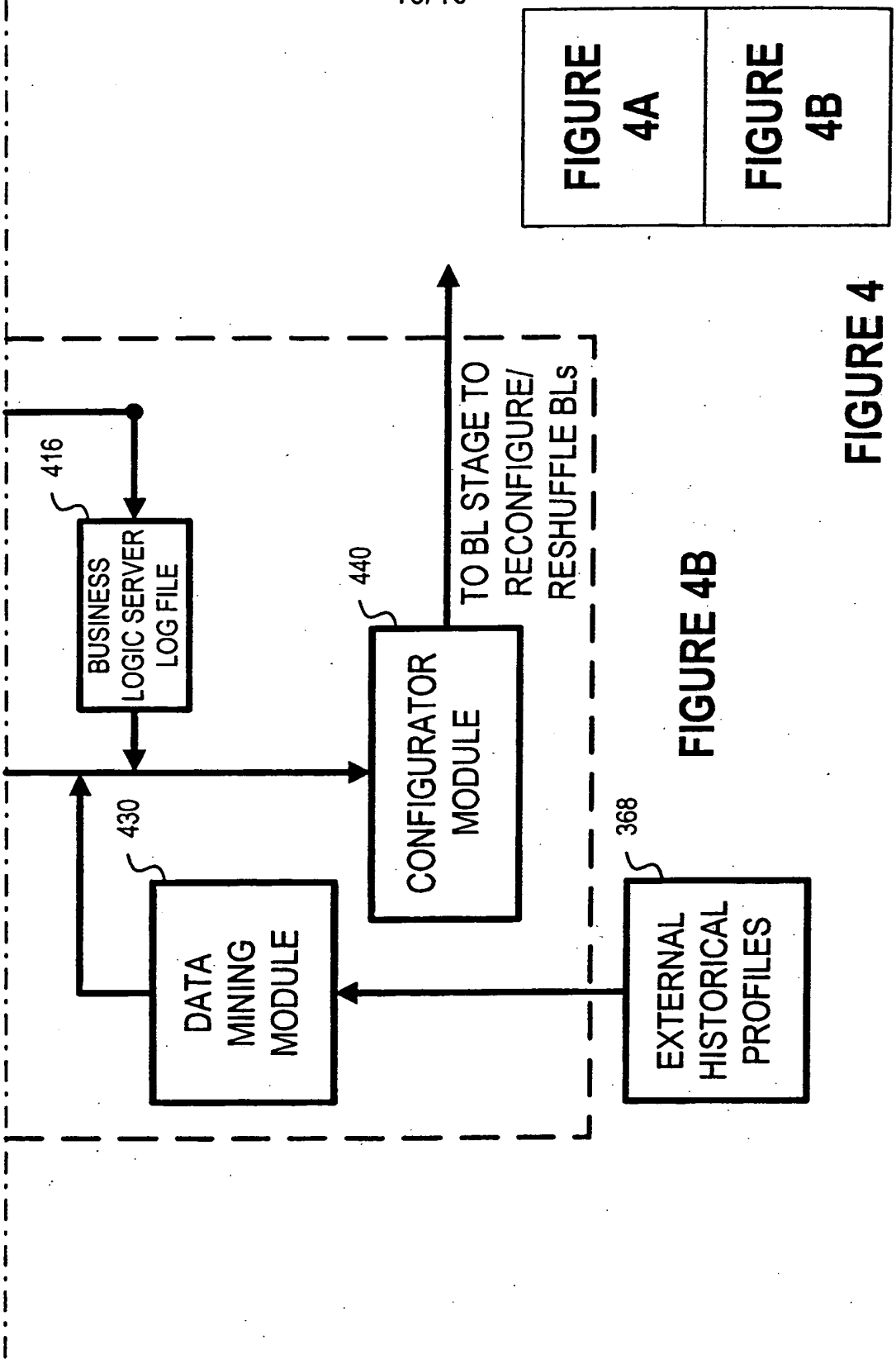


FIGURE 4B

FIGURE 4